<u>**Some Sample AP Computer Science A Questions - Solutions**</u>

Note: These aren't from actual AP tests. I've created these questions based on looking at actual AP tests. Also, in cases where it's not necessary to have choices, I've removed the choices and changed the format of the question to short answer.

<u>**Short Answer Questions**</u>

1) What does the following segment of Java code print out?

```
double x = 4.5;
int y = (int)x;
System.out.println(x+" "+y);
```

<u>**Solution**</u>
This prints out:

```
4.5 4
```

When you cast a double to an int, it truncates the result.

2) What does the following segment of Java code print out?

```
int x = 7, y = 3;
int z = x/y + x%y;
if (z == x)
    y++;
if (z == y)
    x++;
System.out.println(x+" "+y+" "+z);
```

<u>**Solution**</u>
This prints out:

```
8 3 3
```

Since 7/3 = 2 and 7%3 = 1, initially we set z to 3. At this point z and x aren't equal, so the first if gets skipped. Then we evaluate the second if where z and y are equal and x gets incremented to 8 (from 7). So, the final values of x, y and z are 8, 3, and 3 respectively.

3) What does the following segment of Java code print out?

```
for (int i=1; i<30; i = i + 3)
{
    if (i%2 == 0)
        System.out.print((i/2)+" ");
}
System.out.println();
```

<u>Solution</u>
As we go through the loop, i takes on the values 1, 4, 7, 10, 13, 16, 19, 22, 25 and 28. Of these, the even values (those that pass the if statement test) are 4, 10, 16, 22 and 28. If we divide each of these by 2, we get the output:

```
2 5 8 11 14
```

4) What does the following segment of Java code do in general?

```
Scanner stdin = new Scanner(System.in);
System.out.println("Enter n.");
int n = stdin.nextInt();
int total = 0;
while (n > 0) {
    total = total + n%10;
    n = n/10;
}
System.out.println(total);
```

<u>Solution</u>
For any value of n, n%10 isolates the units digit and n/10 chops off the units digit. Thus, the while loop systematically chops off each digit, adding it to total, until no more digits remain. Thus, this segment of code prints out the sum of the digits of the original value of n entered by the user.

Multiple Choice Questions

1) Which of the following Boolean expressions are equivalent? (Assume that x and y are integer variables that have been initialized with the intended values.)

```
i)  ((x > 0) && (y > 0)) || ((x > 0) && (y < 0))

ii) x != y

iii) (x > 0) && (y != 0)

iv) (x > 0) && (x + y != x)
```

A) i and ii
B) i and iii
C) i, iii, iv
D) ii, iii, iv
E) All 4 are different, logically.

Solution
When we look at the first expression, we see that $x > 0$ is a requirement, since this must be true to satisfy either of the larger clauses. Notice that once we force $x > 0$, as long as y isn't 0, the first expression is true, since the or allows either option.

Once we have that down, we can look at the other choices. The second choice isn't equivalent because could just use x = -3, y = 4 to show that (i) and (ii) are different. In this case, the (i) is false while (ii) is true.

Notice that (iii) is exactly what we determined (i) to be.

Finally, notice that x + y != x is equivalent to y != 0, which we can show by subtracting x from both sides of the check. **This, the correct answer is C.**

2) The Boolean expression `!(A || B)` is equivalent to which of the following?

A) `!A || !B`
B) `!A || B`
C) `!A && !B`
D) `!A && B`
E) `None of the Above`

Solution
If we plug in the four options for the ordered pair (A, B) of (True, True), (True, False), (False, True) and (False, False), we see that the given expression is ONLY true for (False, False). Of the ones given, only choice C is ONLY true for (False, False), since it's an AND. **Thus, the correct answer is choice C.** (Note: This is an application of DeMorgan's Law.)

3) What does the following code segment print out?

```
for (int i=5; i>0; i--)
{
    for (int j=0; j<i; j++)
        System.out.print((2*i-j)+" ");
    System.out.println();
}
```

```
A) 5 4 3 2 1   B) 10 9 8 7 6   C) 10 8 6 4 2   D) 0 2 4 6 8
   4 3 2 1         8 7 6 5          8 6 4 2          0 2 4 6
   3 2 1           6 5 4            6 4 2            0 2 4
   2 1             4 3              4 2              0 2
   1               2                2                0
```

E) None of the Above

Solution
As the outer loop runs, i counts down 5, 4, 3, 2, and 1. The value of 2*i in these situations is 10, 8, 6, 4 and 2, respectively. The inner loop counts up from 0 to i-1, so when i is 5, j assumes the values of 0, 1, 2, 3 and 4. Thus, what gets printed on the first line is 10 9 8 7 6. **Tracing through the rest of the lines, we see that choice B is indeed correct.**

1) Write two *different* segments of code that, assuming that the integer variable n has been assigned to a positive value, print out the following:

```
1 2 3 4 … n
1 2 ..n-1
…
1 2
1
```

2) Complete the program below so that it prints out the digits of the integer n, in reverse order. For example, if the user enters 73546, then your program should print out 64537. (Hint: one of the short answer questions may shed light on how to solve this problem!)

```java
import java.util.*;
public class revdigits {

    public static void main(String[] args) {

        Scanner stdin = new Scanner(System.in);
        System.out.println("Enter n.");
        int n = stdin.nextInt();

        // Fill in your code here!

    }
}
```

3) The formula for how far an object falls from a height of h, in t sections is $16t^2$ feet. Thus, the object's height off the ground after t seconds is equal to $max(0, h - 16t^2)$. Complete the program below so that it prints out a chart of how high off the ground an object is at each second (starting at t = 0) until the object hits the ground.

```java
import java.util.*;
public class drop {

    public static void main(String[] args) {

        Scanner stdin = new Scanner(System.in);
        System.out.println("Enter the initial height.");
        int height = stdin.nextInt();

        // Fill in your code here!
    }
}
```